



Τμήμα Ηλεκτρολόγων Μηχανικών  
& Μηχανικών Υπολογιστών

ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΕΛΟΠΟΝΝΗΣΟΥ

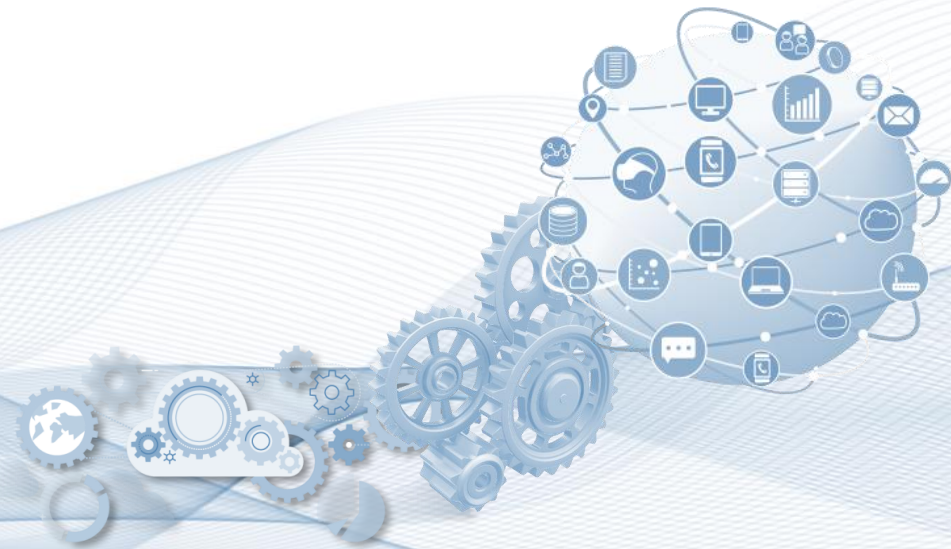
# ΤΕΧΝΙΚΕΣ ΠΡΟΓΡΑΜΜΑΤΙΜΟΥ ΥΠΟΛΟΓΙΣΤΩΝ

## Διάλεξη III

### JavaScript – Μια Εισαγωγή

Γιάννης Τζήμας, Καθηγητής

# JavaScript



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΕΛΟΠΟΝΝΗΣΟΥ  
UNIVERSITY of the PELOPONNESE

# Τι θα δούμε σήμερα...

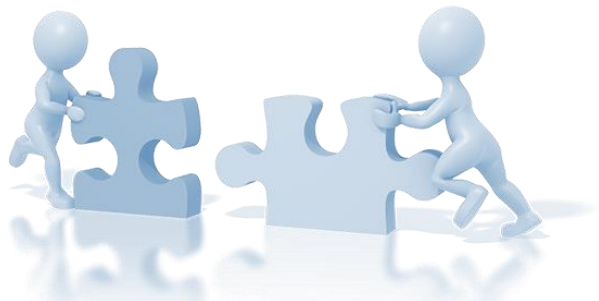
- Βασικός στόχος αυτού του δεύτερου τμήματος του μαθήματος είναι να...
  - προχωρήσουμε **βαθύτερα** στις τεχνικές προγραμματισμού, αξιοποιώντας αυτή τη φορά τη γλώσσα προγραμματισμού **JavaScript**.



# Ας ξαναθυμηθούμε το γιατί είμαστε εδώ!

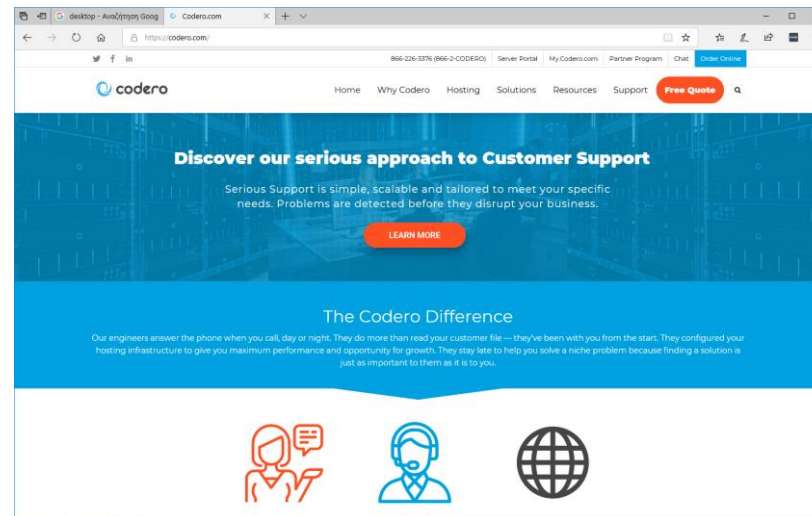
**NOT  
JUST  
ART**

- Όπως ξαναείπαμε ο προγραμματισμός **ανταποδίδει** όσα του δίνετε, είτε αν τον δείτε σαν
  - **χόμπι**, είτε σαν
  - **επάγγελμα**.
- Εμπειριέχει όπως είπαμε **δυνατότητες επίλυσης προβλημάτων (puzzle solving)**.
  - Αυτό πολλοί το θεωρούν **τέχνη!!!**
- Μπορείτε να μαθαίνεταιi κάθε μέρα και κάτι νέο σε σχέση με τον προγραμματισμό και **όσο περισσότερα προγράμματα γράφετε, συνεχώς θα βελτιώνεστε!**



# Ένα από τα πολύ καλά στον προγραμματισμό σήμερα είναι...

- ...το γεγονός ότι υπάρχουν πάρα πολλοί τύποι υπολογιστών και πάρα πολλοί τύποι προγραμμάτων...



# Όπως ξαναείπαμε...



- Όταν δύο άνθρωποι επιλέγουν να **επικοινωνήσουν** διαλέγουν μία **γλώσσα** που και οι δύο καταλαβαίνουν.
- Ο προγραμματισμός είναι κάτι παρόμοιο:
  - Ένα **πρόγραμμα** είναι ένα σύνολο **εντολών** που ο **υπολογιστής** εκτελεί.
  - Οι **εντολές** είναι σε κάποια **γλώσσα** που **καταλαβαίνει** ο υπολογιστής.
  - Καλούμε αυτές τις γλώσσες, **γλώσσες προγραμματισμού**.



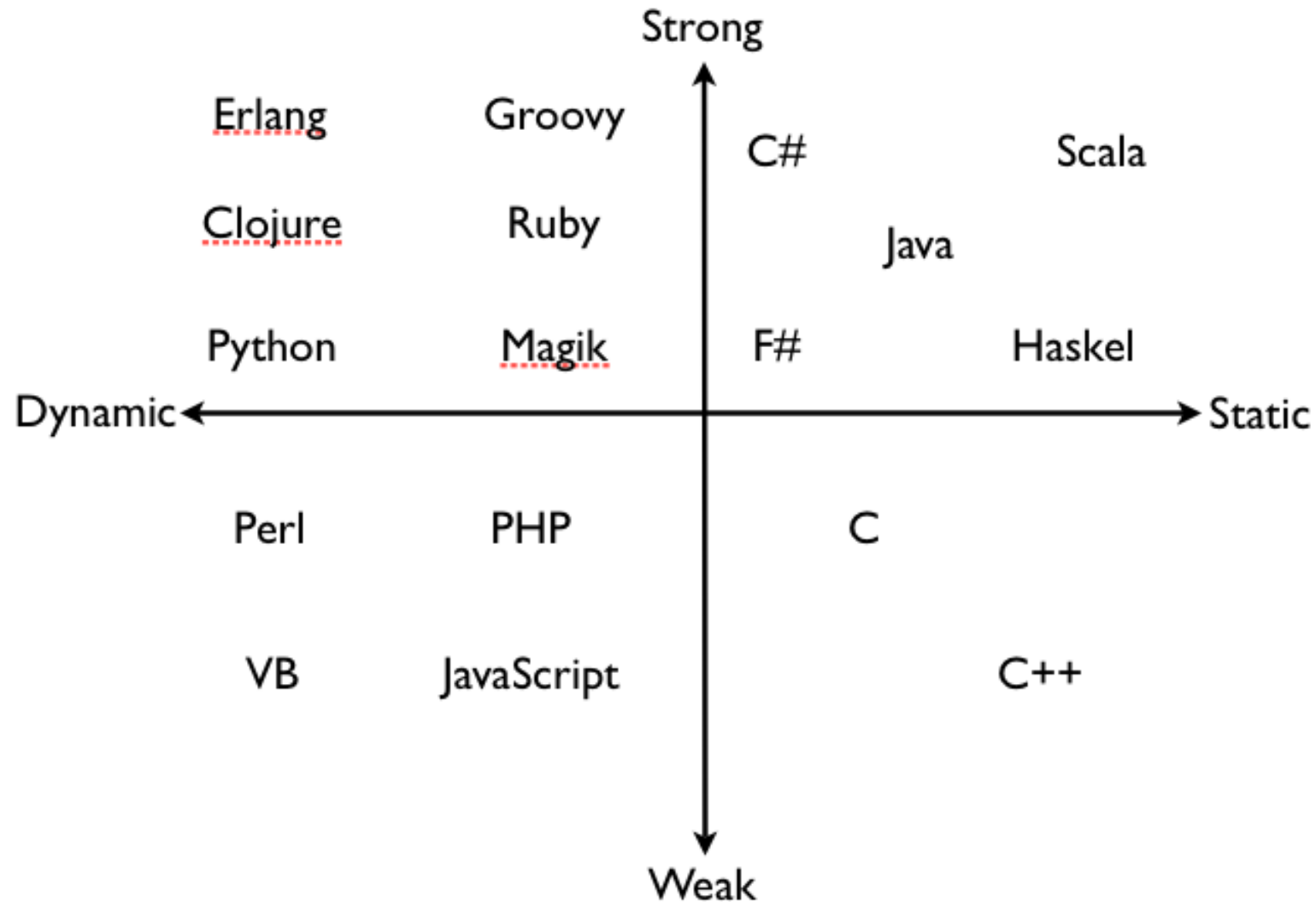
# Γλώσσες Προγραμματισμού...



감사합니다 Natick  
 Danke Ευχαριστίες Dalu  
 Thank You Köszönöm  
 Tack  
 Спасибо Dank Gracias  
 谢谢 Merci Seé  
 ありがとう

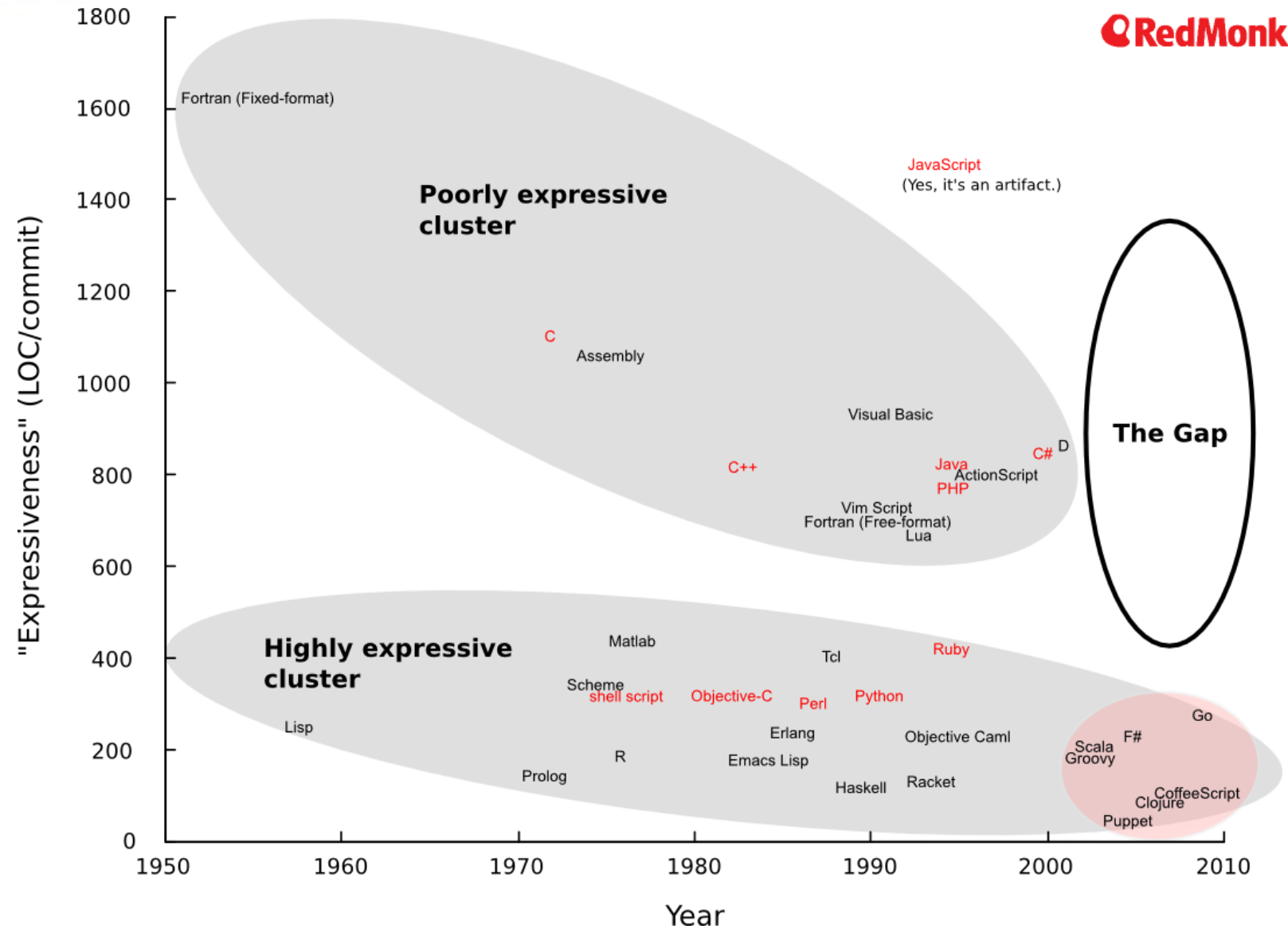
# Γλώσσες Προγραμματισμού...

Πολλά κριτήρια ανάλογα με το τι θέλουμε να πετύχουμε...



# Γλώσσες Προγραμματισμού...

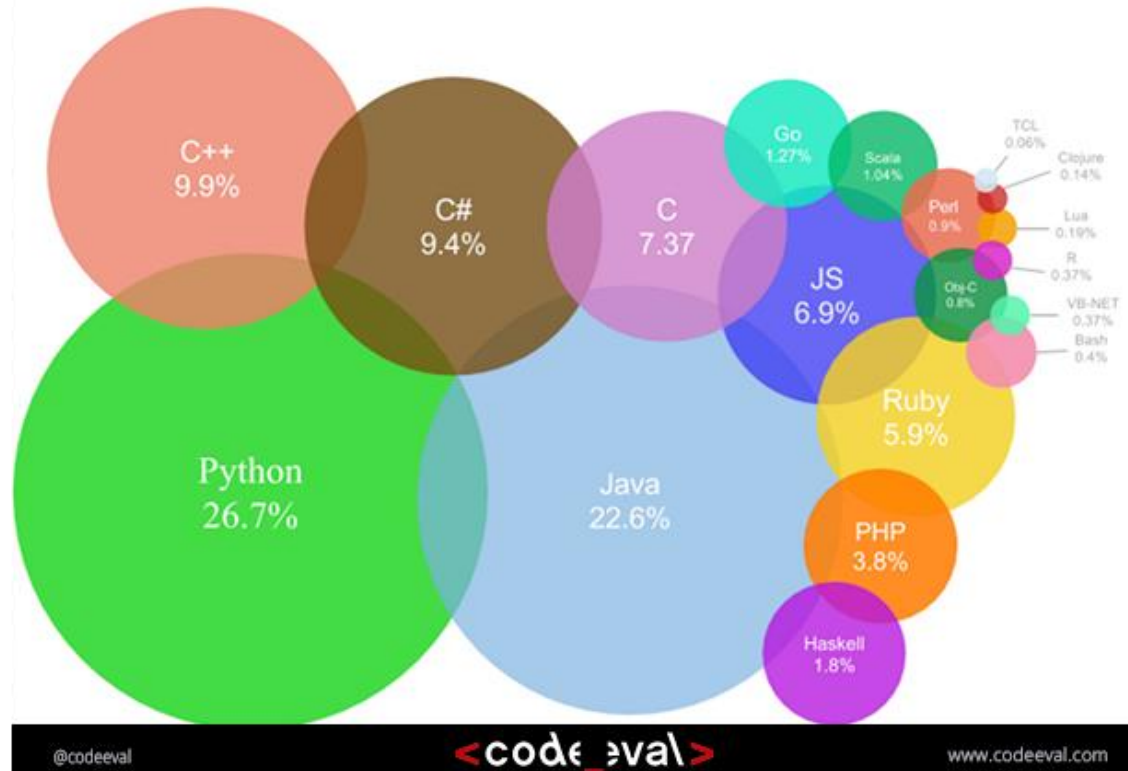
Πολλά κριτήρια ανάλογα με το τι θέλουμε να πετύχουμε...



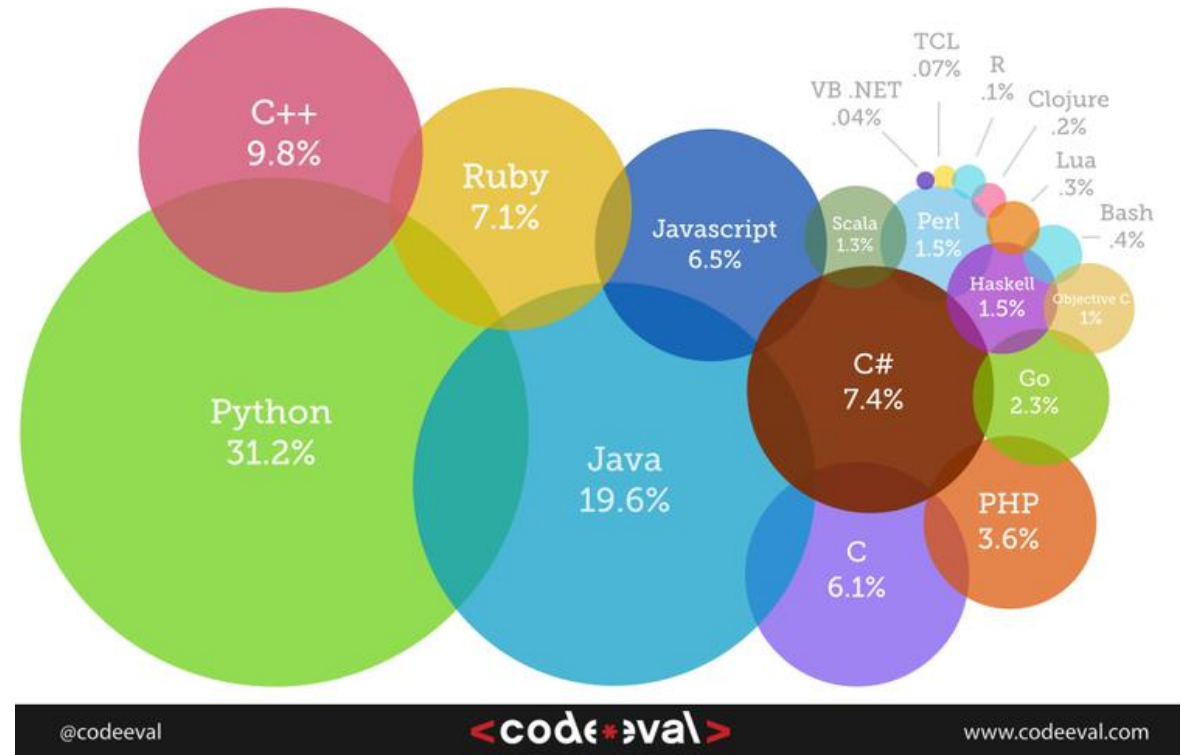


# Γλώσσες Προγραμματισμού... Πόσο Δημοφιλείς είναι;

Most Popular Coding Languages of 2016



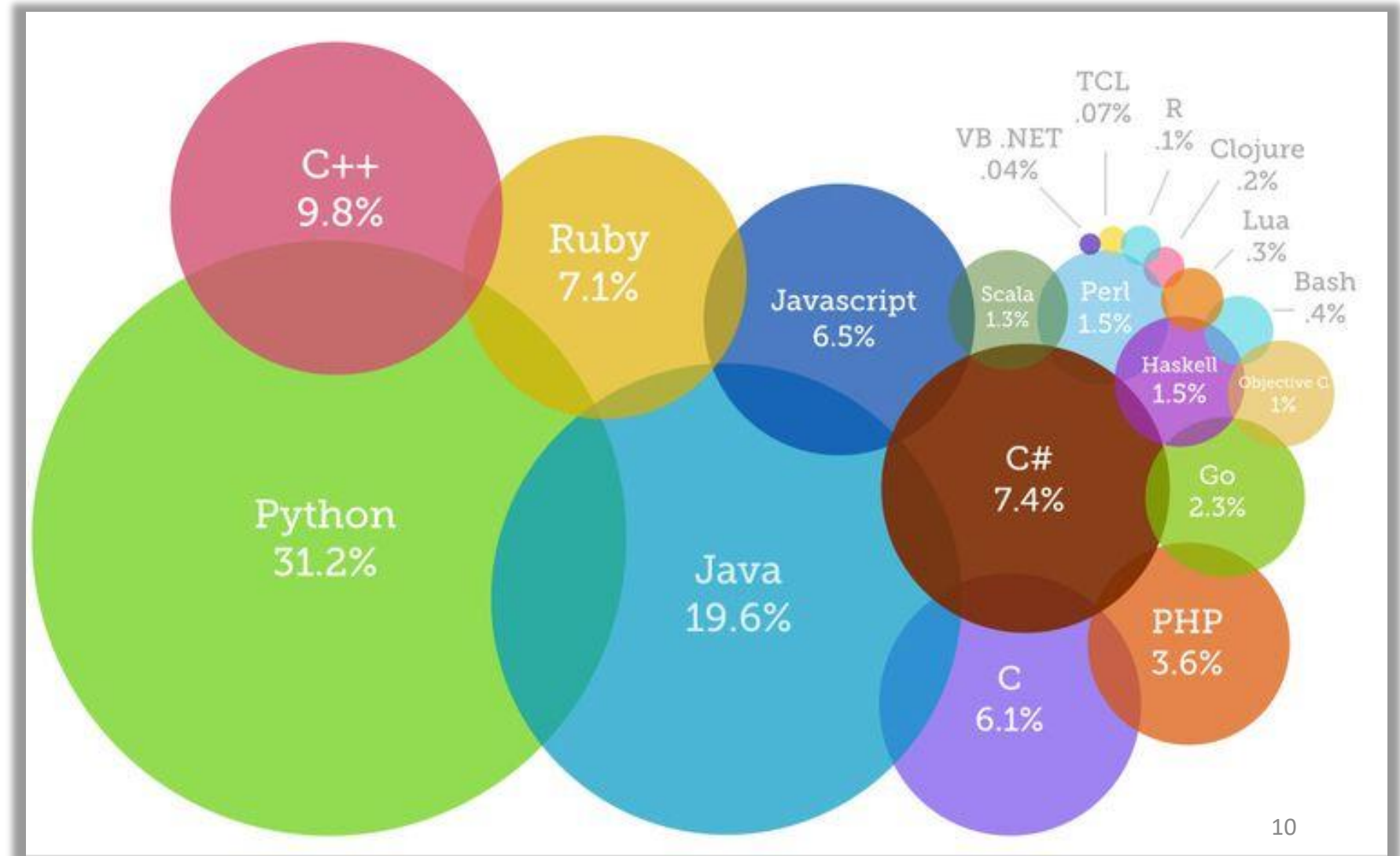
Most Popular Coding Languages of 2015














# Γλώσσες Προγραμματισμού...

## Πόσο Δημοφιλείς είναι;

### 2019



Oct 2024	Oct 2023	Change	Programming Language		Ratings	Change
1	1			Python	21.90%	+7.08%
2	3	▲		C++	11.60%	+0.93%
3	4	▲		Java	10.51%	+1.59%
4	2	▼		C	8.38%	-3.70%
5	5			C#	5.62%	-2.09%
6	6			JavaScript	3.54%	+0.64%
7	7			Visual Basic	2.35%	+0.22%
8	11	▲		Go	2.02%	+0.65%
9	16	▲▲		Fortran	1.80%	+0.78%
10	13	▲		Delphi/Object Pascal	1.68%	+0.38%
11	9	▼		SQL	1.64%	-0.15%
12	14	▲		MATLAB	1.48%	+0.22%
13	20	▲▲		Rust	1.45%	+0.53%
14	12	▼		Scratch	1.41%	+0.05%
15	8	▼▼		PHP	1.21%	-0.69%
16	10	▼▼		Assembly language	1.13%	-0.51%
17	17			R	1.09%	+0.12%
18	19	▲		Ruby	0.99%	+0.07%
19	24	▲▲		COBOL	0.99%	+0.23%
20	15	▼▼		Swift	0.98%	-0.09%

# Γλώσσες Προγραμματισμού... Πόσο Δημοφιλείς είναι; (ΤΙΟΒΕ)

- ΤΙΟΒΕ is specialized in assessing and tracking the quality of software.
- We measure the quality of a software system by applying widely accepted software metrics to it.
- ΤΙΟΒΕ checks more than 1056 million lines of software code for its customers world-wide, real-time, each day.

# Γλώσσες Προγραμματισμού...

## Πόσο Δημοφιλείς είναι; (TIOBE)

Programming Language	2024	2019	2014	2009	2004	1999	1994	1989
Python	1	3	8	7	7	24	22	-
C	2	2	1	2	2	1	1	1
C++	3	4	4	3	3	2	2	2
Java	4	1	2	1	1	4	-	-
C#	5	6	5	6	8	19	-	-
JavaScript	6	7	9	9	10	15	-	-
Visual Basic	7	20	235	-	-	-	-	-
SQL	8	9	-	-	92	-	-	-
Go	9	16	36	-	-	-	-	-
Fortran	10	28	29	26	13	13	5	9
Objective-C	34	10	3	28	39	-	-	-
Lisp	35	32	18	18	14	16	6	3
(Visual) Basic	-	-	6	5	4	3	3	6



# Μία πρώτη γεύση...

## My First JavaScript

Tue Nov 12 2013 11:47:06 GMT+0200 (GTB Standard Time)

Display Date

```
1 <html>
2 <head>
3 <script>
4 function displayDate ()
5 {
6 document.getElementById ("demo") .innerHTML=Date ();
7 }
8 </script>
9 </head>
10 <body>
11
12 <h1>My First JavaScript</h1>
13 <p id="demo">This is a paragraph.</p>
14
15 <button type="button" onclick="displayDate () ">Display Date</button>
16
17 </body>
18 </html>
```

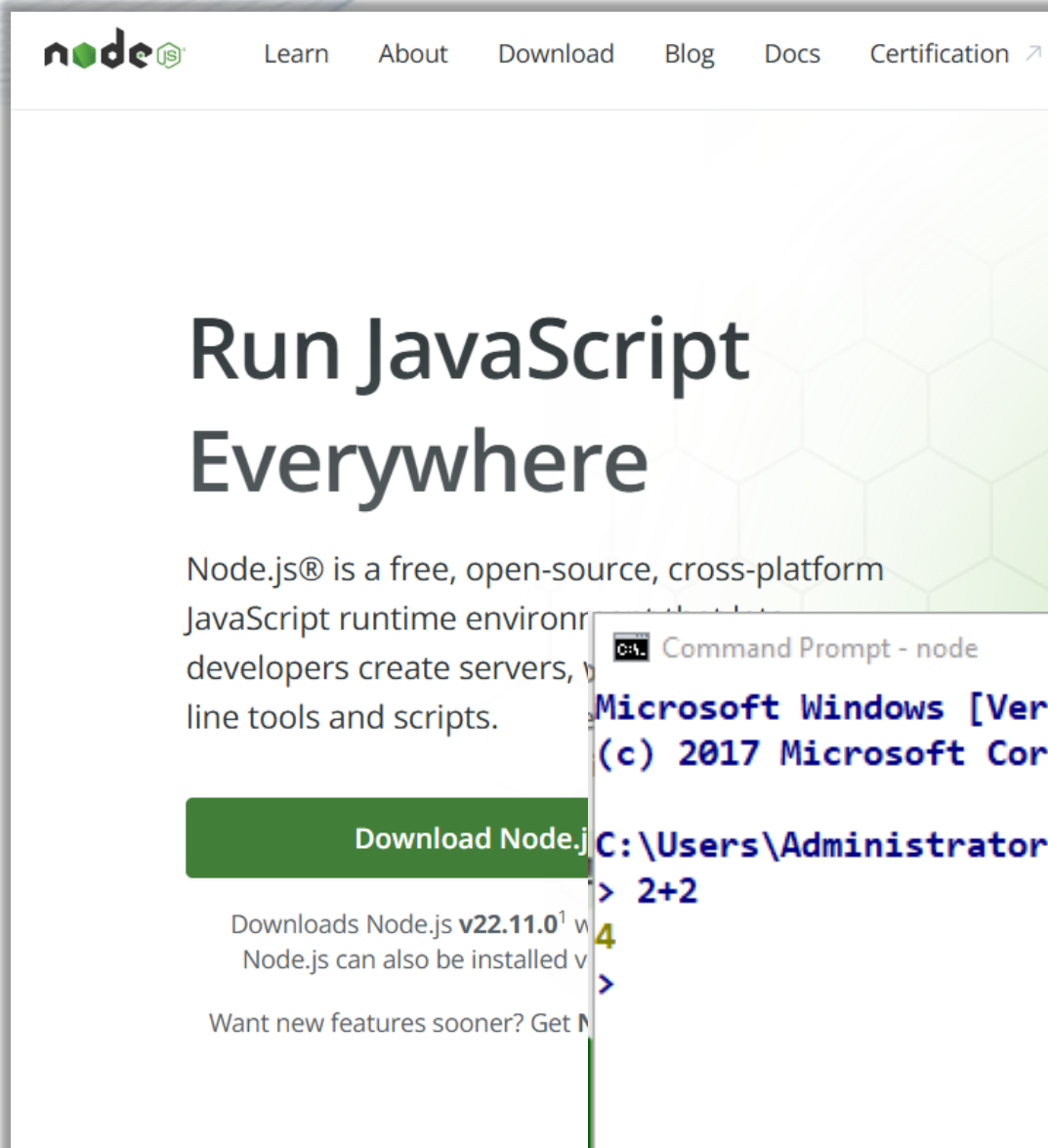
# Και μία ακόμα...



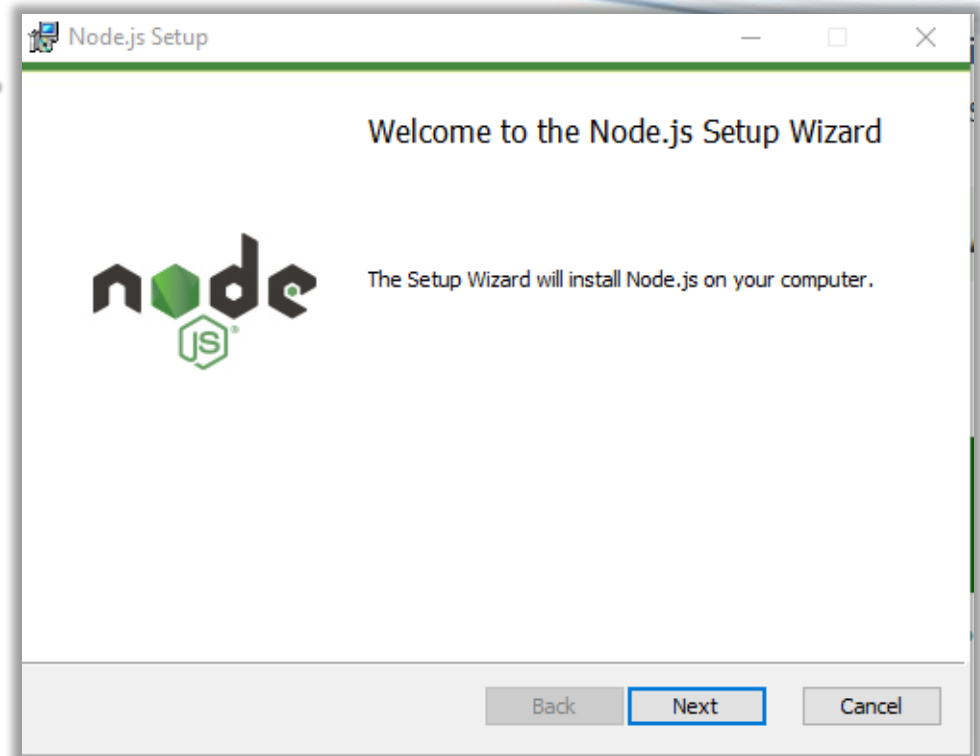
Click the light bulb to turn on/off the light

```
1 <html>
2 <body>
3 <script>
4   function changeImage ()
5   {
6     element=document.getElementById('myimage')
7     if (element.src.match("bulbon"))
8     {
9       element.src="pic_bulboff.gif";
10    }
11    else
12    {
13      element.src="pic_bulbon.gif";
14    }
15  }
16 </script>
17
18 
20
21 <p>Click the light bulb to turn on/off the light</p>
22
23 </body>
24 </html>
```

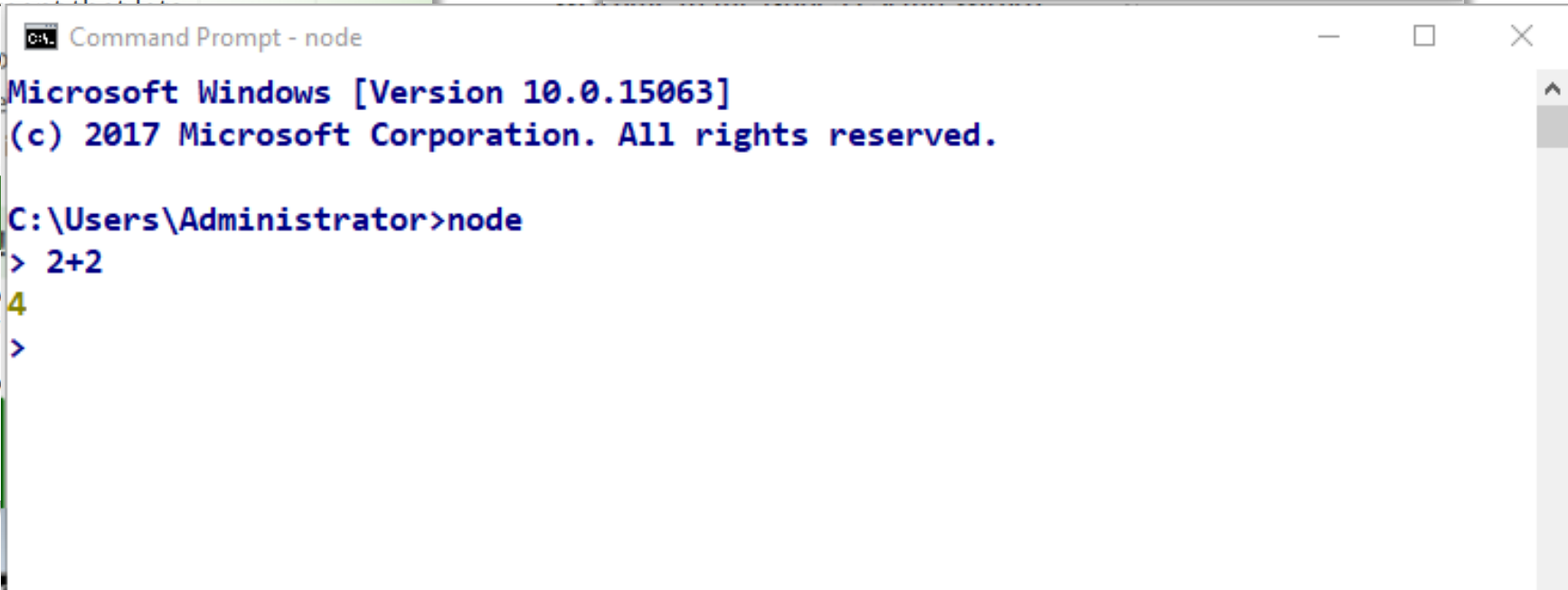
# Node – Εγκαταστήστε το...



The screenshot shows the Node.js website homepage. At the top, there is a navigation menu with links for 'Learn', 'About', 'Download', 'Blog', 'Docs', and 'Certification'. The main heading reads 'Run JavaScript Everywhere'. Below this, a paragraph describes Node.js as a free, open-source, cross-platform JavaScript runtime environment. A prominent green button labeled 'Download Node.js' is visible. Below the button, there is text indicating the current version is v22.11.0 and a link to get new features sooner.



The screenshot shows the 'Node.js Setup' wizard window. It features the Node.js logo and the text 'Welcome to the Node.js Setup Wizard'. A message states: 'The Setup Wizard will install Node.js on your computer.' At the bottom, there are three buttons: 'Back', 'Next' (which is highlighted with a blue border), and 'Cancel'.



The screenshot shows a Windows Command Prompt window titled 'Command Prompt - node'. The text inside the window reads: 'Microsoft Windows [Version 10.0.15063] (c) 2017 Microsoft Corporation. All rights reserved.' The prompt shows the user has entered the command 'C:\Users\Administrator>node' and the output is '> 2+2' followed by a yellow '4' and a '>' character, indicating the command executed successfully.

<http://nodejs.org/>



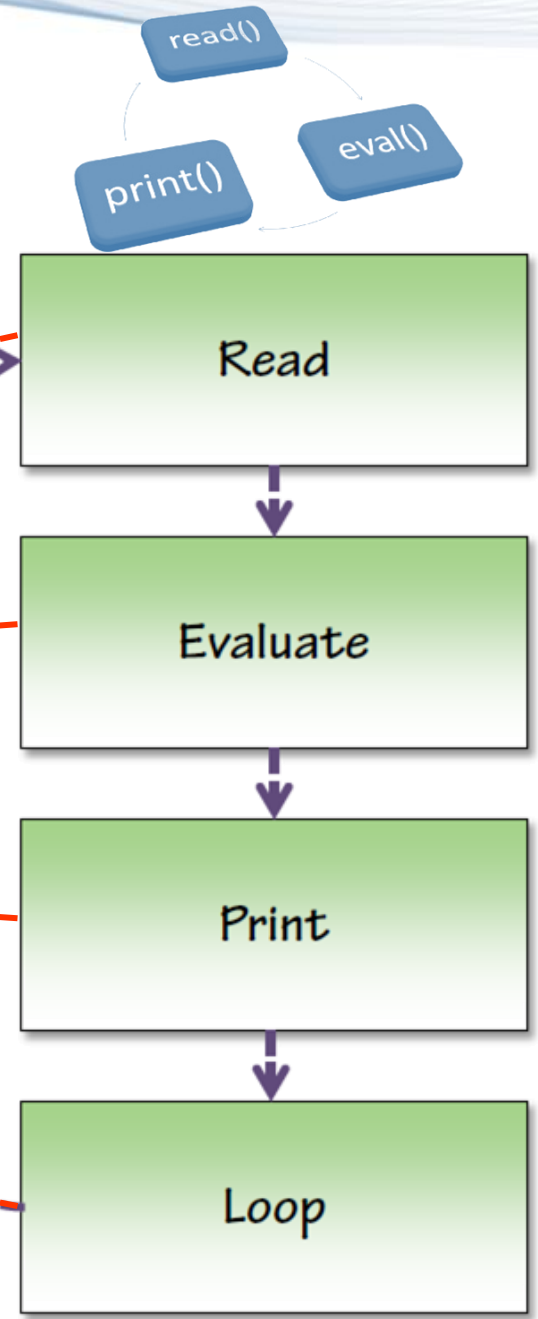
# REPL

## Read-Evaluate-Print-Loop

- Η συμπεριφορά είναι ίδια με την πραγματική ζωή...

```
Command Prompt - node
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>node
> 2+2
4
>
```



# Μερικά πράγματα που μπορούμε να κάνουμε...

```
Command Prompt - node
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>node
> 2+2
4
> 2+3
5
> 2/3
0.6666666666666666
> 2-3
-1
> 3>5
false
> 3<5
true
> 3==5
false
>
```

# ...και που δε μπορούμε...

```
Command Prompt - node
> 3>5
false
> 3<5
true
> 3==5
false
> lunch
ReferenceError: lunch is not defined
    at repl:1:1
    at signHandlersWrap (vm.js:22:35)
    at signHandlersWrap (vm.js:73:12)
    at ContextifyScript.Script.runInThisContext (vm.js:21:12)
    at REPLServer.defaultEval (repl.js:340:29)
    at bound (domain.js:280:14)
    at REPLServer.runBound [as eval] (domain.js:293:12)
    at REPLServer.<anonymous> (repl.js:539:10)
    at emitOne (events.js:101:20)
    at REPLServer.emit (events.js:188:7)
>
```

```
Command Prompt - node
    at emitOne (events.js:101:20)
    at REPLServer.emit (events.js:188:7)
> {
... asdasd
... 123123
... 234234
... }
ReferenceError: asdasd is not defined
    at repl:2:1
    at signHandlersWrap (vm.js:22:35)
    at signHandlersWrap (vm.js:73:12)
    at ContextifyScript.Script.runInThisContext (vm.js:21:12)
    at REPLServer.defaultEval (repl.js:340:29)
    at bound (domain.js:280:14)
    at REPLServer.runBound [as eval] (domain.js:293:12)
    at REPLServer.<anonymous> (repl.js:539:10)
    at emitOne (events.js:101:20)
    at REPLServer.emit (events.js:188:7)
```

```
> {
... .break
>
```

```
> .help
.break    Sometimes you get stuck, this gets you out
.clear    Alias for .break
.editor    Enter editor mode
.exit     Exit the repl
.help     Print this help message
.load     Load JS from a file into the REPL session
.save     Save all evaluated commands in this REPL session to a file
```

# Ας κάνουμε ένα βήμα ακόμα...

```
Command Prompt - node
at REPLServer.<anonymous> (repl.js:539:10)
at emitOne (events.js:101:20)
at REPLServer.emit (events.js:188:7)
> {
... .break
> .help
.break      Sometimes you get stuck, this gets you out
.clear      Alias for .break
.editor     Enter editor mode
.exit       Exit the repl
.help       Print this help message
.load       Load JS from a file into the REPL session
.save       Save all evaluated commands in this REPL session to a fi
> lunch = 'pizza'
'pizza'
> lunch
'pizza'
> lunch = 'πίτα με γύρο'
'πίτα με γύρο'
'αβψ'
> 'abc'
'abc'
> "abc"
'abc'
> 'hello world'
'hello world'
>
```

assign  
αναθέτω

string

Μεταβλητή  
lunch



# Και ας προϋπολογίσουμε ένα βράδυ μας...

```
> lunch = 'πίτσα με 12 κομμάτια'
'πίτσα με 12 κομμάτια'
> sizeOfParty = 4
4
> totalCost = 14.99
14.99
> totalCost / sizeOfParty
3.7475
> averageCost = totalCost / sizeOfParty
3.7475
> averageCost
3.7475
> message = 'Το γεύμα με ' + lunch + ' για ' + sizeOfParty + ' ανθρώπους θα
κοστίσει ' + averageCost
'Το γεύμα με πίτσα με 12 κομμάτια για 4 ανθρώπους θα σας κοστίσει 3.7475
'
```

concatenation

**Μη επιτρεπτό**  
average cost

**Επιτρεπτό**  
averageCost  
lunchOrder  
averageBalance

**Προσοχή!**  
Η JavaScript είναι **Case Sensitive**.  
Προσοχή επίσης στις **συμβάσεις!**

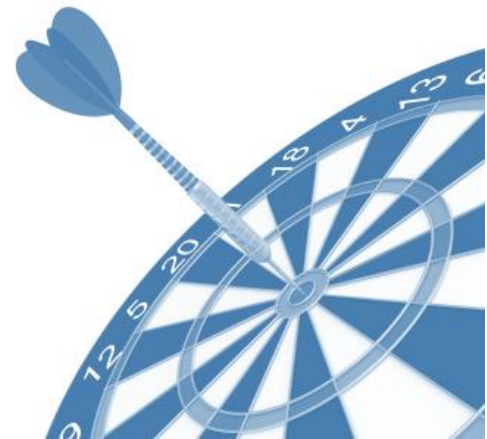
# Τρεις μικρές συμβουλές...

- Δεν υπάρχει δυνατότητα να **μάθετε** προγραμματισμό απλά **διαβάζοντας**.

Πάντα **δοκιμάστε μόνοι σας** όσα βλέπουμε μαζί.

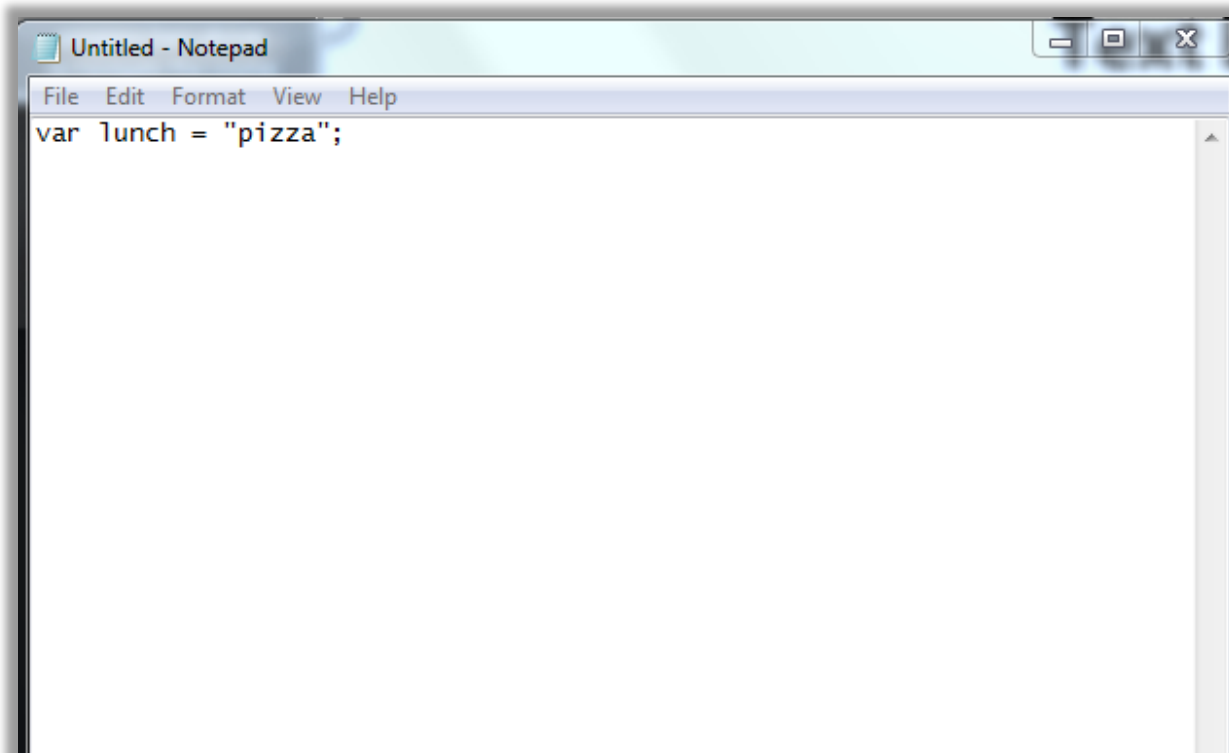
- Πρέπει να είστε πάντα **ακριβείς**.  
Κάνετε διπλούς ελέγχους σε όσα πληκτρολογείτε.

- **Πειραματιστείτε!** Η εμπειρία θα σας κάνει καλύτερους προγραμματιστές!

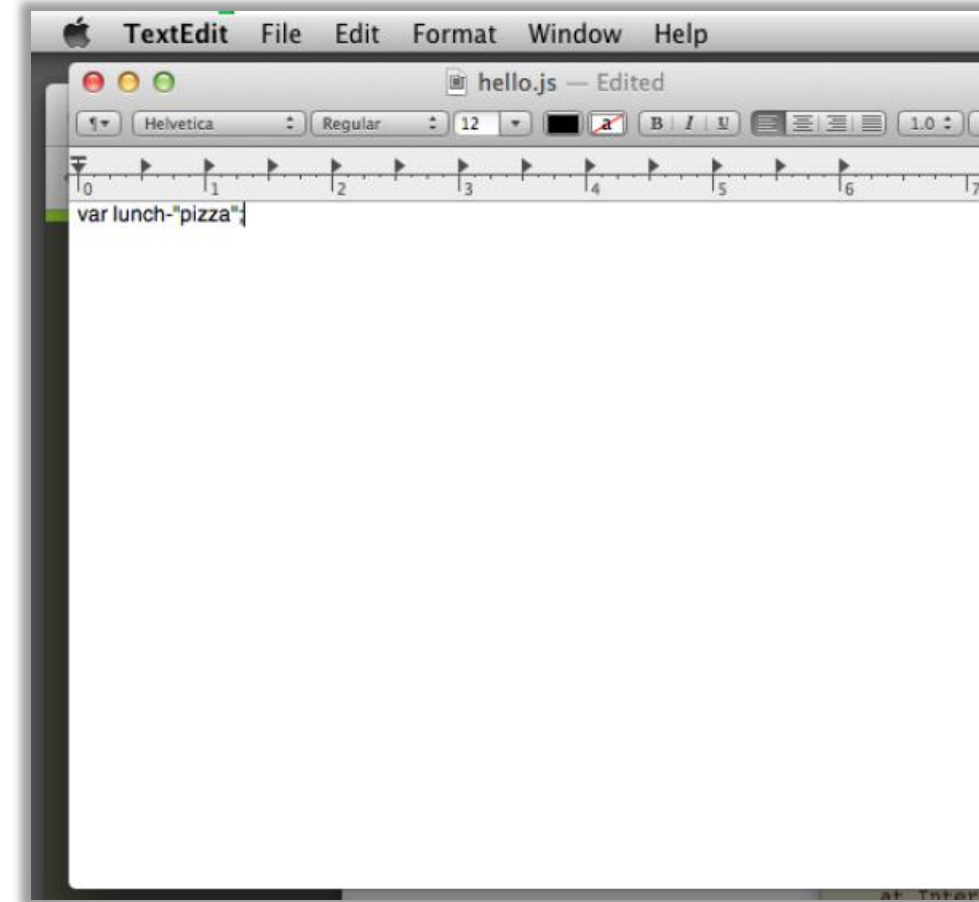


# Και που γράφουμε κώδικα;

- Text Editors
  - Notepad (Windows),
  - TextEdit (Mac),
  - αλλά υπάρχουν και πολύ καλύτερες λύσεις...

A screenshot of a Windows Notepad window titled "Untitled - Notepad". The menu bar includes "File", "Edit", "Format", "View", and "Help". The text area contains the code `var lunch = "pizza";`.

```
var lunch = "pizza";
```

A screenshot of a Mac TextEdit window titled "hello.js — Edited". The menu bar includes "TextEdit", "File", "Edit", "Format", "Window", and "Help". The window shows a ruler at the top and the code `var lunch="pizza";` in the text area.

```
var lunch="pizza";
```

# Δηλαδή...

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.17134.1069]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>notepad

C:\Users\Administrator>cd documents

C:\Users\Administrator\Documents>mkdir examples

C:\Users\Administrator\Documents>cd examples

C:\Users\Administrator\Documents\examples>dir
Volume in drive C has no label.
Volume Serial Number is F4AF-F997

Directory of C:\Users\Administrator\Documents\examples

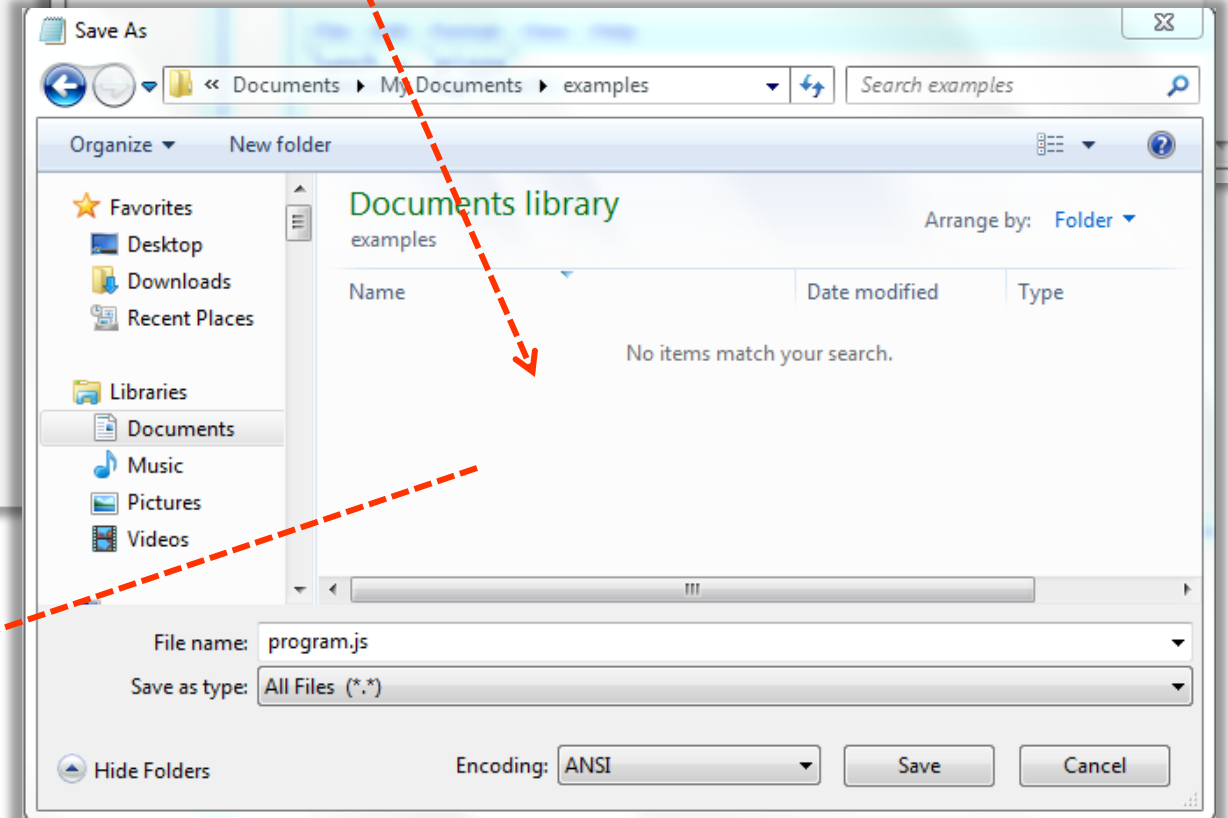
08/11/2019  11:11 πμ    <DIR>          .
08/11/2019  11:11 πμ    <DIR>          ..
                0 File(s)        0 bytes
                2 Dir(s)    171.558.150.144 bytes free

C:\Users\Administrator\Documents\examples>
```

```
C:\Users\Administrator\Documents\examples>node program.js
pizza

C:\Users\Administrator\Documents\examples>
```

```
Untitled - Notepad
File Edit Format View Help
lunch = "pizza"
console.log(lunch)
```





# Πιο έξυπνες λύσεις....

www.yaldex.com

JavaScript Editor Free JS Editor Buy JS Editor Free Scripts Contacts

## Free JavaScript Editor

 Version 4.7

SHARE 

**Free JavaScript Editor** can be used for professional editing *JavaScript code* and creating special effects for webpages using CSS, DHTML, JavaScript and Ajax.

 AJAX developers can use this **free javascript editor** as Ajax Editor>.

*Freeware JavaScript Editor* uses Intellisense to make simpler writing JavaScript code and make it more error-free.

**Free JavaScript Editor** can aid you navigate through JavaScript code using built-in "Functions and Variables" navigator.

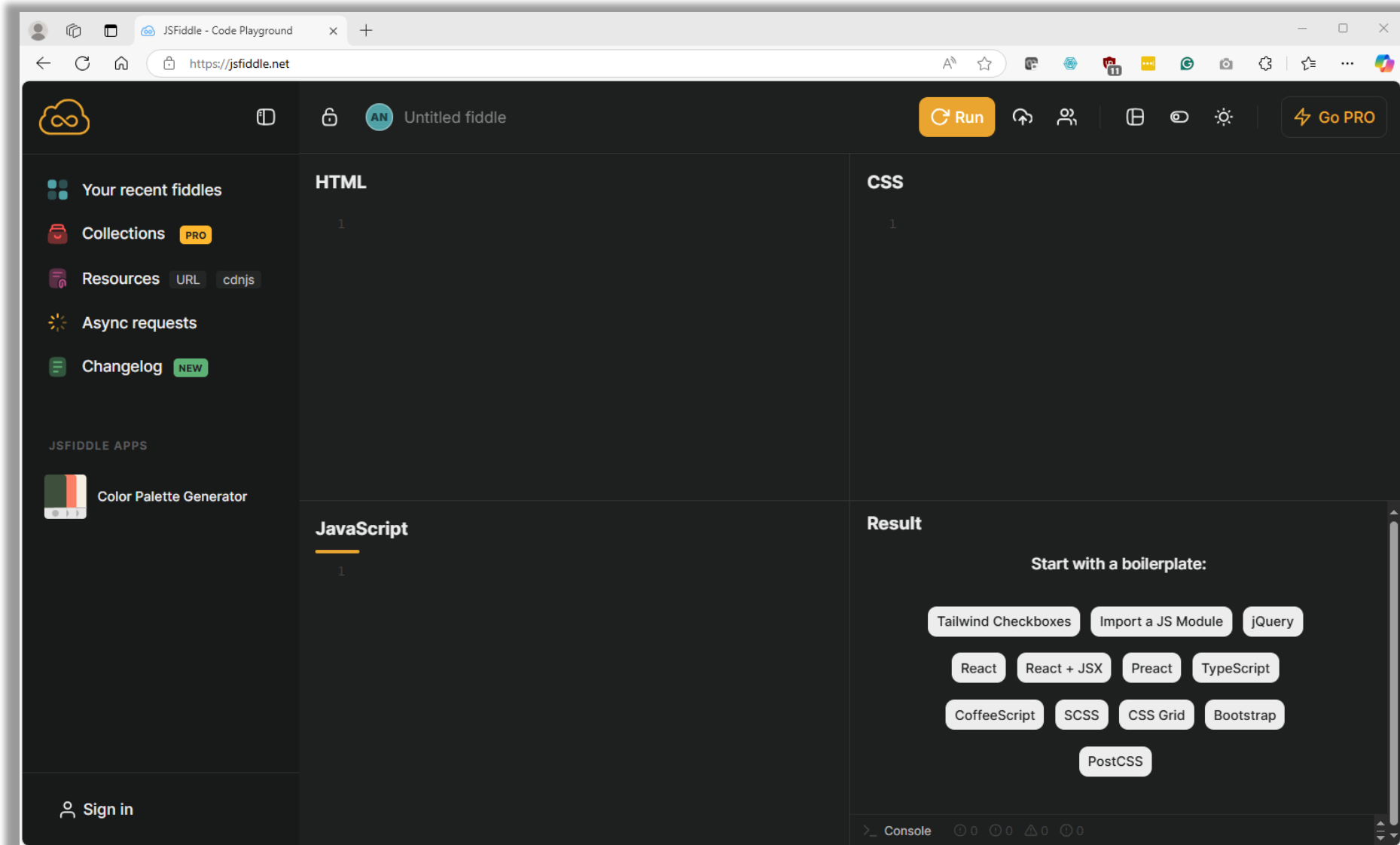
**Note!**  
Intellisense is available during only 21 days in Free JavaScript Editor.  
There are also missing some features in the free version of javascript editor.  
If you want to get all features of the JavaScript Editor you should use [1st JavaScript Editor Pro.](#)

Free JavaScript Editor [C:\debugger.js\*]



# Πιο έξυπνες λύσεις.... (online)

- Και υπάρχουν πολλές ακόμα ψάχνοντας στο Διαδίκτυο...



# Syntax Rules – Συντακτικοί Κανόνες

- Κάθε γλώσσα προγραμματισμού έχει ένα καθορισμένο **συντακτικό** (**syntax**).
- Σε **διαφορετικές γλώσσες** το συντακτικό μπορεί να **διαφέρει αρκετά**.
- Το συντακτικό έχει σχέση με το πώς **διαμορφώνεις** τον κώδικα, ώστε η γλώσσα να μπορεί να τον καταλάβει.

```
var lunch = "pizza"  
console.log(lunch)  
console.log(lunch)  
console.log(4)
```

**Keyword:** πρόκειται για μία λέξη με ξεχωριστή σημασία για τη γλώσσα.

**Βέλτιστη πρακτική:** Καλό είναι να χρησιμοποιείτε τη λέξη **var** όταν δηλώνετε μία μεταβλητή, παρόλο που είναι προαιρετικό.

```
var lunch = "pizza";  
console.log(lunch);  
console.log(lunch);  
console.log(4);
```

Προαιρετικό επίσης...

Ευκολία στην ανάγνωση...

```
var name = "John";  
name = "Alex";
```

```
var name = "John";  
name = "Alex";
```

# Semicolon (;)

- Καλό θα είναι να **χρησιμοποιείται**, γιατί η μη χρήση της μοιάζει με χρήση της ακόλουθης πρότασης:
- *Η Μαρία χώρισε με το αγόρι της. Ο Γιώργος δεν είχε ιδέα ότι ήταν στενοχωρημένη*

αντί για...

- *Η Μαρία χώρισε με το αγόρι της. Ο Γιώργος δεν είχε ιδέα ότι ήταν στενοχωρημένη*

```
var lunch = "pizza"; console.log(lunch);
```

# Αν πρόκειται να κάνετε λάθος, καλύτερα να γίνει νωρίτερα!

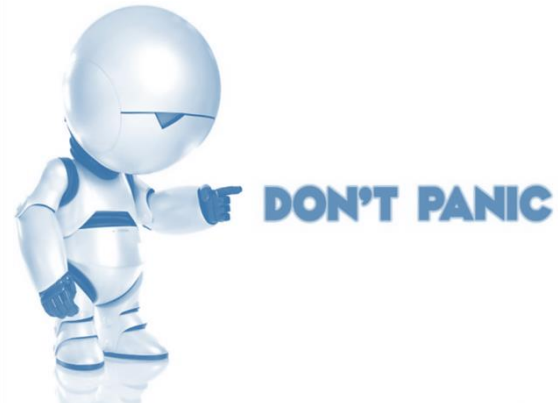
- Η εμπειρία που θα αποκτήσετε στη διάγνωση λαθών είναι ανεκτίμητη, ιδιαίτερα όταν οι γραμμές του κώδικα γίνουν πολλές!

```
var lunch = "pizza" console.log(lunch);
```

```
var lunch = "pizza";  
console.l og(lunch);  
console.log(lunch);  
console.log(4);
```

```
C:\Users\Administrator\Documents\examples>node program.js  
C:\Users\Administrator\Documents\examples\program.js:1  
lunch = "pizza" console.log(lunch);  
                    ^^^^^^^  
  
SyntaxError: Unexpected identifier  
    at Module._compile (internal/modules/cjs/loader.js:892:18)  
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:973:10)  
    at Module.load (internal/modules/cjs/loader.js:812:32)  
    at Function.Module._load (internal/modules/cjs/loader.js:724:14)  
    at Function.Module.runMain (internal/modules/cjs/loader.js:1025:10)  
    at internal/main/run_main_module.js:17:11
```

```
C:\Users\Administrator\Documents\examples\program.js:3  
console.l og(lunch);  
          ^^
```



# Case Sensitive

- Στη JavaScript έχει σημασία η χρήση μικρών και κεφαλαίων γραμμάτων!

```
var Lunch = "pizza";  
  
console.log(lunch);  
console.log(lunch);  
console.log(4);  
  
var name = "John";  
name = "Alex";  
console.log(name);
```

## Challenge Round:

Προπαθήστε να δημιουργήσετε λάθη στο πρόγραμμα!

```
C:\Users\Administrator\Documents\examples\program.js:3  
console.log(lunch);  
           ^  
ReferenceError: lunch is not defined  
    at Object.<anonymous> (C:\Users\Administrator\Documents\examples\program.js:3  
:13)  
[90m    at Module._compile (internal/modules/cjs/loader.js:956:30)[39m  
[90m    at Object.Module._extensions..js (internal/modules/cjs/loader.js:973:10)  
[39m  
[90m    at Module.load (internal/modules/cjs/loader.js:812:32)[39m  
[90m    at Function.Module._load (internal/modules/cjs/loader.js:724:14)[39m  
[90m    at Function.Module.runMain (internal/modules/cjs/loader.js:1025:10)[39m  
[90m    at internal/main/run_main_module.js:17:11[39m
```

# Ας προϋπολογίσουμε ξανά το βράδυ μας...

- `>process.`  
(και πατήστε το πλήκτρο **Tab**)

Το **στιγμιότυπο** ενός προγράμματος που τρέχει.

```
Command Prompt - node

process.chdir
process.cpuUsage
process.debugPort
process.emitWarning
process.execArgv
process.exit
process.hrtime
process.memoryUsage
process.nextTick
process.pid
process.reallyExit
process.stderr
process.stdout
process.umask
process.version

process.config
process.cwd
process.dlopen
process.env
process.execPath
process.features
process.kill
process.moduleLoadList
process.openStdin
process.platform
process.release
process.stdin
process.title
process.uptime
process.versions

> process.title
'Command Prompt - node'
> process.ver
undefined
> process.version
'v6.11.5'
> process.argv
[ 'C:\\Program Files\\nodejs\\node.exe' ]
>
```



# Ας πάρουμε είσοδο από το χρήστη...

```
1 var totalCost = 14.99;  
2  
3 console.log(process.argv);
```

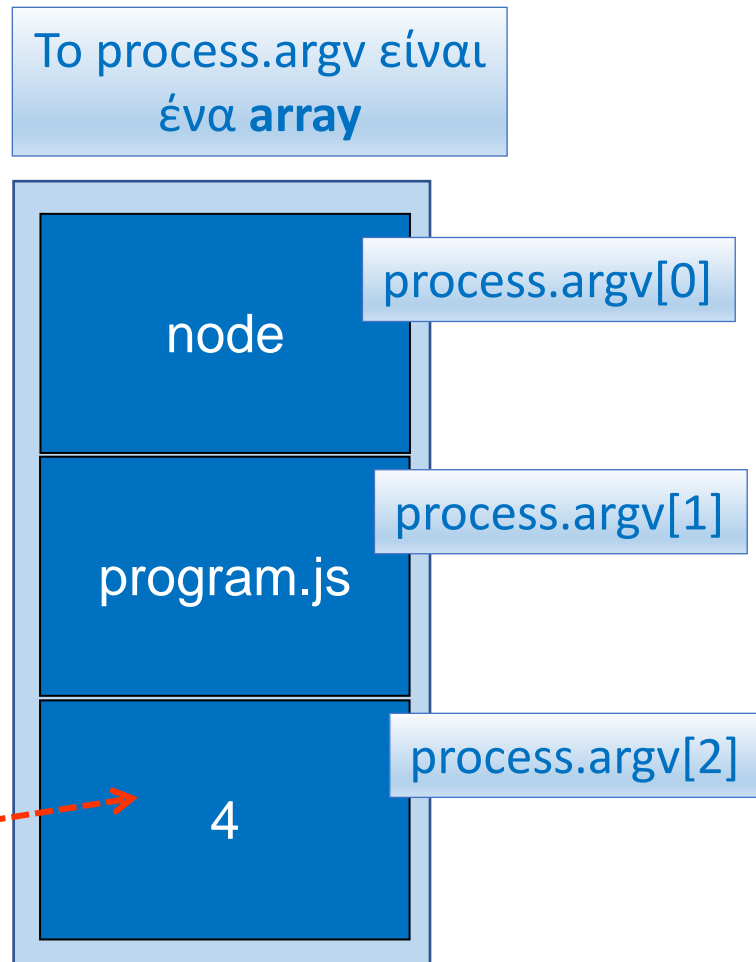
```
C:\Users\Administrator\Documents\examples>node program.js pizza orange onion
```

```
[  
  'C:\\Program Files\\nodejs\\node.exe',  
  'C:\\Users\\Administrator\\Documents\\examples\\program.js',  
  'pizza',  
  'orange',  
  'onion'  
]
```

```
C:\Users\Administrator\Documents\examples>node program.js 4
```


```
C:\Users\Administrator\Documents\examples>node program.js 4  
[  
  'C:\\Program Files\\nodejs\\node.exe',  
  'C:\\Users\\Administrator\\Documents\\examples\\program.js',  
  '4'  
]
```

```
1 var totalCost = 14.99;  
2  
3 var sizeOfParty = process.argv[2];
```



# Και σας δούμε το αποτέλεσμα...

```
1 var totalCost = 14.99;  
2  
3 var sizeOfParty = process.argv[2];  
4  
5 var averageCost = totalCost / sizeOfParty;  
6  
7 console.log("$" + averageCost);
```



```
C:\Users\Administrator\Documents\examples>node program.js 4  
$3.7475
```

```
C:\Users\Administrator\Documents\examples>node program.js 3  
$4.996666666666667
```

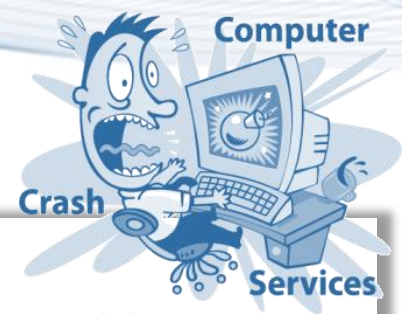
```
C:\Users\Administrator\Documents\examples>node program.js 36  
$0.416388888888889
```

```
C:\Users\Administrator\Documents\examples>node program.js 3610  
$0.004152354570637119
```

# Μεταβλητές & Τύποι δεδομένων

- Σχεδόν όλες οι γλώσσες προγραμματισμού υποστηρίζουν **μεταβλητές (variables)** κάποιου είδους.
  - Είναι ιδιαίτερα **χρήσιμες**.
  - Μας επιτρέπουν να **αποθηκεύουμε δεδομένα** σε κάποιο πρόγραμμα και να τα ανακτούμε μετέπειτα σε κάποιο άλλο χρονικό σημείο, όταν τα χρειαστούμε.
  - Μπορείτε να τις σκέφτεστε σαν τις **μεταβλητές στα μαθηματικά**, αλλά αν δε σας αρέσουν τα μαθηματικά μην ανησυχείτε, εδώ δε θα τα χρησιμοποιήσουμε. Απλά θα δούμε πως λειτουργούν οι μεταβλητές.
- Επίσης, θα δούμε τους **τύπους δεδομένων (Data Types)** στη JavaScript.

# Οι μεταβλητές πιο αναλυτικά



```
1 console.log(x);
```

```
C:\Users\Administrator\Documents\examples>node variables.js
C:\Users\Administrator\Documents\examples\variables.js:1
(function (exports, require, module, __filename, __dirname) { console.log(x);
                                                                    ^
ReferenceError: x is not defined
    at Object.<anonymous> (C:\Users\Administrator\Documents\examples\variables.js:1:75)
    at Module._compile (module.js:570:32)
    at Object.Module._extensions..js (module.js:579:10)
    at Module.load (module.js:487:32)
    at tryModuleLoad (module.js:446:12)
    at Function.Module._load (module.js:438:3)
    at Module.runMain (module.js:604:10)
    at run (bootstrap_node.js:383:7)
    at startup (bootstrap_node.js:149:9)
    at bootstrap_node.js:496:3
```

```
1 var x;
2
3 console.log(x);
```

```
C:\Users\Administrator\Documents\examples>node variables.js
undefined
```

- Οι μεταβλητές θα πρέπει να έχουν μία τιμή. Θα πρέπει να τις **αρχικοποιήσετε**.
- Αποθηκεύουν τα δεδομένα στη **μνήμη** μέχρι το πρόγραμμα να τερματιστεί.

```
1 var x;
2
3 x = 3;
4 console.log(x);
```

# Δήλωση Μεταβλητών – Συμβάσεις

```
1  var x;  
2  
3  x = 3;  
4  console.log(x);  
5  console.log(x);  
6  
7  x = 5;  
8  console.log(x);  
9  
10 var y = 2;  
11  
12 console.log(y);
```



```
1  var x;  
2  var y = 2;  
3  
4  x = 3;  
5  console.log(x);  
6  console.log(x);  
7  
8  x = 5;  
9  console.log(x);  
10  
11 console.log(y);
```



```
1  var x, y = 2;  
2  
3  x = 3;  
4  console.log(x);  
5  console.log(x);  
6  
7  x = 5;  
8  console.log(x);  
9  
10 console.log(y);
```

- Το αποτέλεσμα είναι το ίδιο αλλά είναι πολύ σημαντικό να ακολουθούμε **συμβάσεις** για να είναι πιο **ευανάγνωστος (readable)** ο κώδικάς μας.



```
var x = 3,  
    y = 2;
```



```
var x = 3, y = 2;
```

# Τύποι Δεδομένων – Data Types

- Οι τιμές που αναθέτουμε σε κάποια μεταβλητή έχουν ένα **τύπο** – **τύπο δεδομένων**.
- Ένας τύπος δεδομένων είναι στην πραγματικότητα η **ταξινόμηση** μίας **τιμής**.
- Στις **διάφορες γλώσσες** προγραμματισμού υπάρχουν **διάφοροι τύποι** δεδομένων.

```
1 var x = 3,  
2     y = 2.5;  
3 var z = '3';  
4 var result = x > y;  
5  
6 console.log(typeof x);  
7 console.log(typeof y);  
8 console.log(typeof z);  
9 console.log(result);  
10 console.log(typeof result);  
11 console.log(typeof console);
```

```
C:\Users\Administrator\Documents\examples>node types.js  
number  
number  
string  
true  
boolean  
object
```

# Προσοχή όμως στο ακόλουθο:

```
1 var x = 3,  
2     y = 2.5;  
3 var z = '3';  
4 var result = x > y;  
5  
6 console.log(typeof x);  
7 console.log(typeof y);  
8 console.log(typeof z);  
9  
10 y = "Είμαι αριθμός?";  
11 console.log(typeof y);  
12  
13 console.log(result);  
14 console.log(typeof result);  
15 console.log(typeof console);
```

- **Προσοχή!**  
Οι τύποι δεδομένων είναι τόσο σημαντικοί γιατί ο τύπος από κάτι συχνά συμβάλει στην απόφαση του **τι** μπορείτε να κάνετε με αυτό και πως θα συμπεριφερθεί!

```
C:\Users\Administrator\Documents\examples>node types.js  
number  
number  
string  
string  
true  
boolean  
object
```

# Στην πράξη η JavaScript βοηθάει... αλλά...

```
1 var x = 3,  
2   y = 2.5;  
3 var z = '3';  
4 var result = x > y;  
5  
6 var answer = x + y;  
7  
8 console.log(answer);  
9 console.log(typeof answer);
```

```
C:\Users\Administrator\Documents\examples>node types.js  
5.5  
number
```

```
1 var x = 3,  
2   y = 2.5;  
3 var z = '3';  
4 var result = x > y;  
5  
6 var answer = x + z;  
7  
8 console.log(answer);  
9 console.log(typeof answer);
```

```
C:\Users\Administrator\Documents\examples>node types.js  
33  
string
```

- Η JavaScript προσπαθεί να σας βοηθήσει κάνοντας υποθέσεις, **αλλά...**



# Στην πράξη η JavaScript βοηθάει... αλλά...

```
1 var x = 3,  
2   y = 2.5;  
3 var z = '3';  
4 var result = x > y;  
5  
6 var answer = x * y;  
7  
8 console.log(answer);  
9 console.log(typeof answer);
```

```
C:\Users\Administrator\Documents\examples>node types.js  
7.5  
number
```

```
1 var x = 3,  
2   y = 2.5;  
3 var z = '3';  
4 var result = x > y;  
5  
6 var answer = z * y;  
7  
8 console.log(answer);  
9 console.log(typeof answer);
```

```
C:\Users\Administrator\Documents\examples>node types.js  
7.5  
number
```

- Η JavaScript προσπαθεί να σας βοηθήσει κάνοντας υποθέσεις, **αλλά...**

# Κάνετε μερικές δοκιμές...

- Προσθέστε το 'Hello' με το 'Hello'
- Πολλαπλασιάστε το '2.5' με το '3.5'
- Πολλαπλασιάστε το 'Hello' με το 'Hello'
  
- Αν δείτε “NaN” σημαίνει “not a number”  
(με άλλα λόγια δεν είναι εφικτός ο υπολογισμός).



# Αν θέλαμε όμως να πάρουμε τα πράγματα στα χέρια μας...

```
1 var x = 3,  
2   y = 2.5;  
3 var z = '3';  
4 var result = x > y;  
5  
6 var answer = z + y;  
7  
8 console.log(answer);  
9 console.log(typeof answer);
```



Αν θέλουμε η απάντηση να είναι αριθμός....



```
1 var x = 3,  
2   y = 2.5;  
3 var z = '3';  
4 var result = x > y;  
5  
6 var answer = parseInt(z) + y;  
7  
8 console.log(answer);  
9 console.log(typeof answer);
```

Μπορούμε να  
το κάνουμε!

```
C:\Users\Administrator\Documents\examples>node types.js  
5.5  
number
```

# Και μία μικρή εισαγωγή στις συναρτήσεις πριν τις δούμε αναλυτικά...

- Ας δούμε έναν ακόμα ειδικό τύπο στη JavaScript.

```
1 var x = 3,  
2     y = 2.5;  
3 var z = '3';  
4 var result = x > y;  
5  
6 var answer = parseInt(z) + y;  
7  
8 console.log(answer);  
9 console.log(typeof answer);  
10  
11 console.log(typeof y);  
12 console.log(typeof z);  
13 console.log(typeof result);  
14 console.log(typeof console);  
15 console.log(typeof console.log);
```

```
C:\Users\Administrator\Documents\examples>node types.js  
5.5  
number  
number  
string  
boolean  
object  
function
```

- Οι **συναρτήσεις (functions)** στη Javascript είναι ιδιαίτερα **ισχυρές** γιατί περιέχουν κώδικα που μπορείτε να καλέσετε εντός του προγράμματός σας – αυτό κάναμε και με την **console.log**.

# Συναρτήσεις – Functions

```
1 var writeGreeting = function() {  
2     console.log("Hello!");  
3     console.log("Hello!");  
4 }  
5  
6 writeGreeting();
```

Κώδικας

Κλήση

```
C:\Users\Administrator\Documents\examples>node types.js  
Hello!  
Hello!
```

```
1 var writeGreeting = function() {  
2     var message = "Hello!";  
3     console.log(message);  
4     console.log(message);  
5 }
```

Στις συναρτήσεις μπορούμε να ορίσουμε μεταβλητές

```
1 var writeGreeting = function(name) {  
2     var message = "Hello! " + name;  
3     console.log(message);  
4 }  
5  
6 writeGreeting("John");  
7 writeGreeting("Mary");
```

Στις συναρτήσεις μπορούμε να περάσουμε παραμέτρους

```
C:\Users\Administrator\Documents\examples>node types.js  
Hello! John  
Hello! Mary
```

- Κάντε μια μικρή δοκιμή. Τι συμβαίνει αν χρησιμοποιήσετε παρενθέσεις δίπλα σε μία μεταβλητή που δεν είναι συνάρτηση; π.χ. x();

# Τι είδαμε μέχρι τώρα;

- Κάναμε μία **εισαγωγή** στη JavaScript.
- Είδαμε **εργαλεία** με τα οποία μπορούμε να **γράψουμε** κώδικα σε JavaScript και πως να τον **τρέξουμε**.
- Ασχοληθήκαμε με κάποια εισαγωγικά σημεία σε σχέση με το **συντακτικό** της γλώσσας.
- Μιλήσαμε για τις **μεταβλητές** και τους **τύπους δεδομένων** που υποστηρίζει η JavaScript και τέλος...
- Είδαμε πολύ επιφανειακά στις **συναρτήσεις** με τις οποίες θα ασχοληθούμε αναλυτικά στο επόμενο μάθημα.



# JavaScript

